# Claims

What is claimed is:

[c1] A method for transparently injecting a proxy into a distributed application having a server portion and a client portion, comprising:

creating the proxy that implements an interface of a remote object in the server portion and has a capability to cache data from the remote object;

modifying the client portion to substitute a call for the remote object with a call for the proxy; and

interposing a client runtime that includes the proxy between the client portion and the server portion.

[c2] The method of claim **1**, further comprising:

interposing a server runtime between the client runtime and the server portion that enables interaction between the client runtime and the server portion.

[c3] The method of claim **2**, further comprising:

sending a message to the server runtime to fetch data from the remote object.

[c4] The method of claim **3**, further comprising:

receiving data fetched by the server runtime and caching data in the proxy.

[c5] The method of claim **4**, further comprising:

sending a message to the server runtime to synchronize data cached in the proxy with data in the remote object.

[c6] The method of claim **2**, further comprising:

sending a message to the server runtime to invoke a method of the remote object on behalf of the proxy.

**[c7]** The method of claim **6**, further comprising:

receiving a result of invoking the method of the remote object from the server runtime and passing the result to the proxy.

**[c8]** A method for transparently injecting a proxy into a distributed application having a server portion and a client portion, comprising:

creating the proxy for a plurality of remote objects in the server portion, each proxy implementing an interface of a corresponding remote object and having a capability to cache data from the corresponding remote object;

modifying the client portion to substitute a call for a remote object with a call for a corresponding proxy; and

interposing a runtime that includes the proxy between the client portion and the server portion.

**[c9]** The method of claim **8**, wherein creating the proxy for the plurality of remote objects in the server portion comprises analyzing the server portion to determine the remote objects in the server portion.

**[c10]** The method of claim **9**, wherein analyzing the server portion comprises parsing machine code for the server portion.

**[c11]** The method of claim **9**, wherein analyzing the server portion comprises parsing a descriptor containing a list of classes in the server portion.

**[c12]** The method of claim **9**, wherein analyzing the server portion comprises parsing source code for the server portion.

**[c13]** The method of claim **8**, wherein modifying the client portion comprises modifying machine code for the client portion.

**[c14]** The method of claim **8**, wherein modifying the client portion comprises modifying source code for the client portion.

**[c15]** The method of claim **8**, further comprising:
modifying the client portion to substitute a call to a first lookup service that locates the remote object with a call to a second lookup service that locates the corresponding proxy.

**[c16]** The method of claim **15**, wherein the lookup service that locates the corresponding proxy is included in the runtime.

**[c17]** The method of claim **8**, further comprising:
modifying the client portion to substitute a call to manage a lifecycle of the remote object with a call to manage a lifecycle of the corresponding proxy.

**[c18]** The method of claim **8**, further comprising:
fetching data from the remote object into the proxy associated with the remote object.

**[c19]** The method of claim **18**, further comprising:
synchronizing data in the proxy with data in the remote object associated with the proxy.

**[c20]** The method of claim **8**, further comprising:
invoking a method of the remote object on behalf of the proxy associated with the remote object.

**[c21]** The method of claim **20**, further comprising:
receiving a result of invoking the method of the remote object and passing the result to the proxy.

**[c22]** The method of claim **8**, wherein the runtime includes a client runtime that interacts with the client portion and a server runtime that interacts with the server portion.

**[c23]** The method of claim **22**, wherein the client runtime and server runtime communicate in order to enable interaction between the client portion and the server portion.

**[c24]** A method for transparently injecting a proxy into a distributed application having a server portion and a client portion, comprising:

analyzing the server portion to find each remote object in the server portion;

creating the proxy for each remote object in the server portion and including the proxy in a runtime library;

analyzing the client portion to determine calls made to remote objects in the server portion and replacing calls for remote objects with calls for a corresponding proxy; and

interposing the runtime library between the client portion and the server portion.

**[c25]** A method for optimizing a distributed application having a server portion and a client portion, comprising:

interposing a runtime between the client portion and the server portion, the runtime comprising at least a proxy associated with a remote object in the server portion, the proxy having a capability to cache state information from the remote object;

enabling the client portion to interact with the proxy;

fetching data from the remote object into the proxy; and

synchronizing data in the proxy with data in the remote object.

**[c26]** A computer-readable medium having recorded thereon instructions executable by a processor, the instructions for:

generating a proxy for a remote object in a server and making the proxy available to a client;

fetching data from the remote object into the proxy; and

returning data in the proxy to the remote object.

**[c27]** The computer-readable medium of claim **26**, wherein returning data in the proxy to the remote object comprises returning data modified by the client to the remote object.

**[c28]** The computer-readable medium of claim **26**, further comprising:

instructions for updating data in the remote object with data in the proxy.

**[c29]** The computer-readable medium of claim **26**, further comprising:
instructions for invoking a method of the remote object on behalf of the proxy.

**[c30]** The computer-readable medium of claim **29**, further comprising:
instructions for returning a result of invoking the method of the remote object to the proxy.

**[c31]** A computer-readable medium having recorded thereon instructions executable by a processor, the instructions for:
analyzing a server portion of a distributed application to find each remote object in the server portion;
generating a proxy for each remote object in the server portion; and
including the proxy for each remote object in the server portion in a runtime library.

**[c32]** The computer-readable medium of claim **31**, further comprising:
instructions for modifying a client portion of the distributed application such that a call for a remote object is replaced with a call for a corresponding proxy.

**[c33]** An optimizer for a distributed application, comprising:
means for creating a local proxy for a remote object in the distributed application;
means for fetching data from the remote object into the local proxy; and
means for synchronizing data in the local proxy with data in the remote object.

**[c34]** The optimizer of claim **33**, further comprising:
means for invoking a method of the remote object on behalf of the local proxy.

**[c35]** The optimizer of claim **34**, further comprising:
means for receiving a result of invoking the method of the remote object and passing the result to the local proxy.

**[c36]** The optimizer of claim **33**, further comprising:
means for packing data for transport between the local proxy and the remote object.